A Convex Surrogate Operator for General Non-Modular Loss Functions

Jiaqian Yu, Matthew Blaschko

A Convex Surrogate Operator for General Non-Modular Loss Functions

Jiaqian Yu
Inria & CentraleSupélec, Université Paris-Saclay
Grande Voie des Vignes
92295 Châtenay-Malabry, France
jiaqian.yu@centralesupelec.fr

Matthew B. Blaschko
Center for Processing Speech and Images
Departement Elektrotechniek, KU Leuven
3001 Leuven, Belgium
matthew.blaschko@esat.kuleuven.be

Abstract

Empirical risk minimization frequently employs convex surrogates to underlying discrete loss functions in order to achieve computational tractability during optimization. However, classical convex surrogates can only tightly bound modular loss functions, submodular functions or supermodular functions separately while maintaining polynomial time computation. In this work, a novel generic convex surrogate for general non-modular loss functions is introduced, which provides for the first time a tractable solution for loss functions that are neither supermodular nor submodular. This convex surrogate is based on a submodular-supermodular decomposition for which the existence and uniqueness is proven in this paper. It takes the sum of two convex surrogates that separately bound the supermodular component and the submodular component using slack-rescaling and the Lovász hinge, respectively. It is further proven that this surrogate is convex, piecewise linear, an extension of the loss function, and for which subgradient computation is polynomial time. Empirical results are reported on a non-submodular loss based on the Sørensen-Dice difference function, and a real-world face track dataset with tens of thousands of frames, demonstrating the improved performance, efficiency, and scalability of the novel convex surrogate.

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51.

1 Introduction

Many learning problems involve the simultaneous prediction of multiple labels. A simple strategy is to empirically minimize the Hamming loss over the set of predictions [27]. However, this does not always reflect the underlying risk of the prediction process, and may lead to suboptimal performance. Following the risk minimization principle [29], we may instead wish to minimize a loss function that more closely reflects the cost of a specific set of predictions. Alternatives to the Hamming loss are frequently employed in the discriminative learning literature: [4] uses a rank loss which is supermodular; [21] uses a non-submodular loss based on F-score; [6] uses modular losses e.g. Hamming loss and F1 loss which is non-submodular; and losses that are nonmodular are common in a wide range of problems, including Jaccard index based losses [2, 9, 20], or more general submodular-supermodular objectives [19].

This has motivated us to study the conditions for a loss function to be tractably upper bounded with a tight convex surrogate. For this, we make use of the discrete optimization literature, and in particular submodular analysis [11, 24]. Existing polynomial-time convex surrogates exist for supermodular [28] or submodular losses [30], but not for more general non-modular losses. We may perform approximate inference in polynomial time via a greedy optimization procedure to compute a subgradient or cutting plane of a convex surrogate for a general increasing function, but this leads to poor performance of the training procedure in practice [10, 14]. A decomposition-based method for a general set function has been proposed in the literature [13], showing that under certain conditions a decomposition into a submodular plus a supermodular function can be efficiently found. Other relevant work includes the hardness results on submodular Hamming optimization and its approximation algorithms [12].

In this paper, we propose a novel convex surrogate for general non-modular loss functions, which is solvable for the first time for non-supermodular and non-submodular loss functions. In Section 2, we introduce the basic concepts used in this paper. In Section 3, we define a decomposition for a general non-modular loss function into supermodular and submodular components (Section 3.1), propose a novel convex surrogate operator based on this decomposition (Section 3.2), and demonstrate that it is convex, piecewise linear, an extension of the loss function, and for which subgradient computation is polynomial time (Section 3.3). In Section 4, we introduce the Sørensen-Dice loss, which is neither submodular nor supermodular. In Section 5 we demonstrate the feasibility, efficiency and scalability of our convex surrogate with the Sørensen-Dice loss on a synthetic problem, and a range of non-modular losses on a real-world face-track dataset comprising tens of thousands of video frames.

2 Non-modular loss functions

In empirical risk minimization for a set of binary predictions, we wish to minimize some functional of

$$\hat{\mathcal{R}}(h) = \frac{1}{n}\sum_{i=1}^{n}\Delta(y_i, \text{sign}(h(x_i))). \tag{1}$$

For an arbitrary loss function $\Delta : \{-1,+1\}^p \times \{-1,+1\}^p \mapsto \mathbb{R}$, we define a convex surrogate with an operator $\mathbf{B}$,

$$\mathbf{B}\Delta : \{-1,+1\}^p \times \mathbb{R}^p \mapsto \mathbb{R}. \tag{2}$$

We may then minimize the empirical expectation of $\mathbf{B}\Delta(y, h(x))$ with respect to functions $h : \mathcal{X} \mapsto \mathbb{R}^p$. For well behaved function classes for h, minimization of the convex surrogate becomes tractable, provided that subgradient computation of $\mathbf{B}\Delta$ is efficiently solvable.

Any loss function Δ of this form may be interpreted as a set function where inclusion in a set is defined by a corresponding prediction being incorrect:

$$\Delta(y, \tilde{y}) = l(\{i | y^i \neq \tilde{y}^i\}) \tag{3}$$

for some set function l.

In our analysis of convex surrogates for non-modular loss functions, we will employ several results for the Structured Output SVM [28], which assumes that a structured prediction is made by taking an inner product of a feature representation of inputs and outputs: $\text{sign}(h(x)) = \arg\max_y \langle w, \phi(x, y)\rangle$. The slack rescaling variant of the Structured Output SVM is as follows:

$$\min_{w,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i, \quad \forall i, \forall \tilde{y} \in \mathcal{Y} : \tag{4}$$

$$\langle w, \phi(x_i, y_i)\rangle - \langle w, \phi(x_i, \tilde{y})\rangle \geq 1 - \frac{\xi_i}{\Delta(y_i, \tilde{y})} \tag{5}$$

In the sequel, we consider a feature function such that $\langle w, \phi(x, y)\rangle = \sum_{j=1}^{p}\langle w^j, x^j\rangle y^j$. Each w^j is then a vector of length d, and $w \in \mathbb{R}^{d \cdot p}$. Therefore p individual prediction functions parametrized by w^j are simultaneously optimized, although we may also consider cases in which we constrain $w^j = w^i \ \forall i, j$. More generally, we may consider $h : \mathcal{X} \mapsto \mathbb{R}^p$, which may have non-linearities, e.g. deep neural networks.

2.1 Mathematical preliminaries

Definition 1. *A set function l maps from the powerset of some base set V to the reals $l : \mathcal{P}(V) \mapsto \mathbb{R}$.*

Definition 2. *A set function l is non-negative if $l(A) - l(\emptyset) \geq 0, \ \forall A \subseteq V$.*

We denote the set of all such loss functions satisfying Equation (3) $\mathcal{F}$. Following standard conventions in submodular analysis, we assume that $l(\emptyset) = 0$. In this paper we consider l is non-negative, which we will denote $l \in \mathcal{F}_+$.

Definition 3 (Submodular function)**.** *A set function $l : \mathcal{P}(V) \mapsto \mathbb{R}$ is submodular iff for all $B \subseteq A \subset V$ and $x \in V \setminus A$,*

$$l(B \cup \{x\}) - l(B) \geq l(A \cup \{x\}) - l(A) \tag{6}$$

A function is *supermodular* iff its negative is submodular, and a function is modular (e.g. Hamming loss) iff it is both submodular and supermodular. We denote the set of all submodular functions as $\mathcal{S}$, and the set of all supermodular functions as $\mathcal{G}$.

Definition 4. *A set function l is symmetric if $l(A) = c(|A|)$ for some function $c : \mathbb{Z}^* \mapsto \mathbb{R}$.*

Proposition 1. *A symmetric set function l is submodular iff c is concave [1, Proposition 6.1].*

Definition 5 (Increasing function)**.** *A set function $l : \mathcal{P}(V) \mapsto \mathbb{R}$ is increasing if and only if for all subsets $A \subset V$ and elements $x \in V \setminus A$, $l(A) \leq l(A \cup \{x\})$.*

We note that the set of increasing supermodular functions is identical to $\mathcal{G}_+$. We will propose a convex surrogate operator for a general non-negative loss function, based on the fact that set functions can always be expressed as the sum of a submodular function and a supermodular function:

Proposition 2. *For all set functions l, there always exists a decomposition into the sum of a submodular function $f \in \mathcal{S}$ and a supermodular function $g \in \mathcal{G}$:*

$$l = f + g \tag{7}$$

A proof of this proposition is given in [19, Lemma 4].

Proposition 3. *For an arbitrary decomposition $l = f + g$ where g is not increasing, there exists a modular function m_g s.t.*

$$l = (f - m_g) + (g + m_g) \tag{8}$$

with $\tilde{f} := f - m_g \in \mathcal{S}$, and $\tilde{g} := g + m_g \in \mathcal{G}_+$ is increasing.

Proof. Any modular function can be written as

$$m_g(A) = \sum_{j \in A} w_j \tag{9}$$

for some coefficient vector $w \in \mathbb{R}^{|V|}$. For each $j \in V$, we may set

$$w_j = -\min_{A \subseteq V} g(A \cup \{j\}) - g(A). \tag{10}$$

The resulting modular function will ensure that $g + m_g$ is increasing following Definition 5. □

This proof indicates that a decomposition $l = f + g$ is not-unique due to a modular factor. We subsequently demonstrate that decompositions can vary by more than a modular factor:

Proposition 4 (Non-uniqueness of decomposition up to modular transformations.)**.** *For any set function, there exist multiple decompositions into submodular and supermodular components such that these components differ by more than a modular factor:*

$$\exists f_1, f_2 \in \mathcal{S},\ g_1, g_2 \in \mathcal{G}$$
$$(l = f_1 + g_1 = f_2 + g_2) \wedge (g_1 + m_{g_1} \neq g_2 + m_{g_2}) \tag{11}$$

where $\wedge$ denotes "logical and," m_{g_1} and m_{g_2} are constructed as in Equations (9) *and* (10)*.*

Proof. Let m be a submodular function that is not modular. For a given decomposition $l = f_1 + g_1$, we may construct $f_2 := f_1 + m$ and $g_2 := g_1 - m$. As m is not modular, there is no modular m_1 such that $g_1 - m_1 = g_1 - m = g_2$. □

3 A convex surrogate for general non-modular losses

We will show in this section the unique decomposition for a general non-negative loss starting from any arbitrary submodular-supermodular decomposition, which allows us to define a convex surrogate operator based on such a canonical decomposition.

3.1 A canonical decomposition

In this section, we define an operator $\mathbf{D}$ such that $g^* := \mathbf{D}l \in \mathcal{G}_+$ is unique and $f^* := l - \mathbf{D}l \in \mathcal{S}$ is then unique. We have demonstrated in the previous section that we may consider there to be two sources of non-uniqueness in the decomposition $l = f + g$: a modular component and a non-modular component related to the curvature of g (respectively f). We define $\mathbf{D}$ such that these two sources of non-uniqueness are resolved using a canonical decomposition $l = f^* + g^*$.

Definition 6. *We define an operator $\mathbf{D} : \mathcal{F} \mapsto \mathcal{G}_+$ as*

$$\mathbf{D}l = \arg\min_{g \in \mathcal{G}_+} \sum_{A \subseteq V} g(A), \quad \textit{s.t. } l - g \in \mathcal{S}. \tag{12}$$

We note that minimizing the values of g will simultaneously remove the non-uniqueness due both to the modular non-uniqueness described in Proposition 3, as well as the non-modular non-uniqueness described in Proposition 4. We formally prove this in Proposition 5.

Proposition 5. $\mathbf{D}l$ *is unique for all $l \in \mathcal{F}$ that have a finite base set V.*

Proof. We note that the arg min in Equation (12) is equivalent to a linear program: g is uniquely determined by a vector in $\mathbb{R}^{2^{|V|}-1}$ the coefficients of which correspond to $g(A)$ for all $A \in \mathcal{P}(V) \setminus \emptyset$, and we wish to minimize the sum of the entries subject to a set of linear constraints enforcing supermodularity of g, non-negativity of g, and submodularity of $l - g$.

From [18, Theorem 2], an LP of the form

$$\min_{x \in \mathbb{R}^d} \quad r^T x \tag{13}$$
$$\text{s.t.} \quad Cx \geq q \tag{14}$$

has a unique solution if there is no $y \in \mathbb{R}^d$ simultaneously satisfying

$$C_J y \geq 0, \quad r^T y \leq 0, \quad y \neq 0 \tag{15}$$

where $J = \{i | C_i x^* = q_i\}$ is the active set of constraints at an optimum x^*. We note that as r is a vector of all ones (cf. Equation (12)), $r^T y \leq 0$ constrains y to lie in the non-positive orthant. However, as the linear program is minimizing the sum of x subject to lower bounds on each entry of x (e.g. positivity constraints), we know that $C_J y \geq 0$ will bound y to lie in the non-negative orthant. This means, at most, these constraints overlap at $y = 0$, but this is expressly forbidden by the last condition in Equation (15). □

Although Equation (12) is a linear programming problem, we do not consider this definition to be constructive in general as the size of the problem is exponential

in $|V|$ (see [13]). However, it may be possible to verify that a given decomposition satisfies this definition for some loss functions of interest. Furthermore, for some classes of set functions, the LP has lower complexity, e.g. for symmetric set functions the resulting LP is of linear size, and loss functions that depend only on the number of false positives and false negatives (such as the Sørensen-Dice loss discussed in Section 4) result in a LP of quadratic size.

We finally note that from Equation (3), for every $\Delta(y, \cdot)$ we may consider its equivalence to a set function $l = g^* + f^*$, and denote the resulting decomposition of

$$\Delta(y, \cdot) = \Delta_{\mathcal{G}}(y, \cdot) + \Delta_{\mathcal{S}}(y, \cdot) \tag{16}$$

into its supermodular and submodular components, respectively.[1]

3.2 Definition of the convex surrogate

Now that we have defined a unique decomposition $l = g^* + f^*$, we will use this decomposition to construct a surrogate $\mathbf{B}\Delta$ that is convex, piecewise linear, an extension of Δ, and for which subgradient computation is polynomial time. We construct a surrogate $\mathbf{B}$ by taking the sum of two convex surrogates applied to $\Delta_{\mathcal{G}}$ and $\Delta_{\mathcal{S}}$ independently. These surrogates are slack-rescaling [28] applied to $\Delta_{\mathcal{G}}$ and the Lovász hinge [30] applied to $\Delta_{\mathcal{S}}$.

Definition 7 (Slack-rescaling operator [30])**.** *The slack-rescaling operator* $\mathbf{S}$ *is defined as:*

$$\mathbf{S}\Delta(y, h(x)) := \max_{\tilde{y} \in \mathcal{Y}} \Delta(y, \tilde{y}) \left(1 + \langle h(x), \tilde{y} \rangle - \langle h(x), y \rangle\right). \tag{17}$$

The Lovász hinge of a submodular function builds on the Lovász extension [17]:

Definition 8 (Lovász hinge [30])**.** *The Lovász hinge,* $\mathbf{L}$*, is defined as the unique operator such that, for a submodular set function* l *related to* Δ *as in Eq.* (3)*:*

$$\mathbf{L}\Delta(y, h(x)) := \left(\max_{\pi} \sum_{j=1}^{p} s^{\pi_j} \left(l\left(\{\pi_1, \cdots, \pi_j\}\right) - l\left(\{\pi_1, \cdots, \pi_{j-1}\}\right)\right)\right)_+ \tag{18}$$

where $(\cdot)_+ = \max(\cdot, 0)$*,* π *is a permutation,*

$$s^{\pi_j} = 1 - h^{\pi_j}(x) y^{\pi_j}, \tag{19}$$

and $h^{\pi_j}(x)$ *is the* π_j*th dimension of* $h(x)$*.*

[1] Note that Δ_S and Δ_G are due to Eq. (3) for f^* and g^* which explicitly depend on $\mathbf{D}$. For simplicity of notation, we will use $\Delta_{\mathcal{G}}$ instead of $\Delta_{\mathbf{D}\mathcal{G}}$

Algorithm 1 Cutting plane algorithm

```
Input: (x_1, y_1), ..., (x_n, y_n), C, ε
S^i = ∅, ∀i = 1, ..., n
repeat
    for i = 1, ..., n do
    ŷ_L = arg max_ỹ H_L(y_i) = arg max_ỹ LΔ_S
    ŷ_S = arg max_ỹ H_S(y_i) = arg max_ỹ SΔ_G
    H(ŷ) = H_L(ŷ_L) + H_S(ŷ_S)
    ξ^i = max{0, H(y_i)}
    if H(ŷ) > ξ^i + ε then
        S^i := S^i ∪ {y_i}
        w ← optimize Equation (4) with constraints
        defined by ∪_i S^i
    end if
until no S^i has changed during an iteration
return (w, ξ)
```

Definition 9 (General non-modular convex surrogate)**.** *For an arbitrary non-negative loss function* Δ*, we define*

$$\mathbf{B_D}\Delta := \mathbf{L}\Delta_{\mathcal{S}} + \mathbf{S}\Delta_{\mathcal{G}} \tag{20}$$

where $\Delta_{\mathcal{S}}$ *and* $\Delta_{\mathcal{G}}$ *are as in Equation* (16)*, and* $\mathbf{D}$ *is the decomposition of* l *defined by Definition 6.*

We use a cutting plane algorithm to solve the max-margin problem as shown in Algorithm 1.

3.3 Properties of $\mathbf{B_D}$

In the remainder of this section, we show that $\mathbf{B_D}$ has many desirable properties. Specifically, we show that $\mathbf{B_D}$ is closer to the convex closure of the loss function than slack rescaling and that it generalizes the Lovász hinge (Theorems 1 and 2). Furthermore, we formally show that $\mathbf{B_D}\Delta$ is convex (Theorem 3), an extenstion of Δ for a general class of loss functions (Theorem 4), and polynomial time computable (Theorem 5).

Lemma 1. *If* $l \in \mathcal{G}$*, then* $f^* := l - \mathbf{D}l \in \mathcal{S} \cap \mathcal{G}$ *i.e. modular.*

Proof. First we set $l = g_m + f_m$ where f_m is modular and $f_m(\{j\}) = l(\{j\})$. Then any subset $S \subseteq V$

$$f_m(S) = \sum_{j \in S} l(\{j\}),\ g_m(S) = l(S) - \sum_{j \in S} l(\{j\})$$

$$\sum_{S \subseteq V} g_m(S) = \sum_{S \subseteq V} \left(l(S) - \sum_{j \in S} l(\{j\}) \right). \tag{21}$$

The sum in Equation (21) is precisely the sum that should be minimized in Equation (12). We now show that this sum cannot be minimized further while allowing f_m to be non-modular. If there exists any g s.t.

$f := l - g$ is submodular but not modular, by definition there exists at least one subset $S_s \subseteq V$ and one $j \in S_s$ such that

$$f(S_s \setminus \{j\}) + f(\{j\}) > f(S_s) + f(\emptyset). \tag{22}$$

Then by subtracting each time one element from the subset S_s, we have

$$\begin{aligned} f(S_s) &< f(S_s \setminus \{j\}) + f(\{j\}) \\ &\leq f(S_s \setminus (\{j\} \cup \{k\})) + f(\{k\}) + f(\{j\}) \\ &\leq \cdots \leq \sum_{j \in S_s} f(\{j\})), \qquad \forall k \in S_s \setminus (\{j\} \end{aligned} \tag{23}$$

which implies

$$g(S_s) > l(S_s) - \sum_{j \in S_s} l(\{j\}) \tag{24}$$

By taking the sum of the inequalities as in Equation 24 for all subsets S, we have that

$$\sum_{S \subseteq V} g(S) > \sum_{S \subseteq V} \left(l(S) - \sum_{j \in S} l(\{j\}) \right) = \sum_{S \subseteq V} g_m(S)$$

which means $\sum_{S \subseteq V} g(S) > \sum_{S \subseteq V} g_m(S)$ for any g. By Definition 6, $g^* = g_m = \mathbf{D}l$, thus $f^* := l - \mathbf{D}l = f_m$ is modular. □

Lemma 2. *For a loss function Δ such that $\Delta_{\mathcal{S}}$ is increasing, we have*

$$\mathbf{S}\Delta_{\mathcal{G}} = \mathbf{S}(\Delta - \Delta_{\mathcal{S}}) = \mathbf{S}\Delta - \mathbf{S}\Delta_{\mathcal{S}}. \tag{25}$$

Proof. By Definition 7, for every single cutting plane determined by some $\tilde{y}$, we have

$$\begin{aligned} &\mathbf{S}\left(\Delta(y, \tilde{y}) - \Delta_{\mathcal{S}}(y, \tilde{y})\right) \\ &= \left(\Delta(y, \tilde{y}) - \Delta_{\mathcal{S}}(y, \tilde{y})\right)\left(1 + \langle h(x), \tilde{y}\rangle - \langle h(x), y\rangle\right) \\ &= \Delta(y, \tilde{y})\left(1 + \langle h(x), \tilde{y}\rangle - \langle h(x), y\rangle\right) \\ &\quad - \Delta_{\mathcal{S}}(y, \tilde{y})\left(1 + \langle h(x), \tilde{y}\rangle - \langle h(x), y\rangle\right) \\ &= \mathbf{S}\Delta(y, \tilde{y}) - \mathbf{S}\Delta_{\mathcal{S}}(y, \tilde{y}). \end{aligned} \tag{26}$$

As this property holds for all cutting planes, it also holds for the supporting hyperplanes that define the convex surrogate and $\mathbf{S}(\Delta - \Delta_{\mathcal{S}}) = \mathbf{S}\Delta - \mathbf{S}\Delta_{\mathcal{S}}$. □

Definition 10. *A convex surrogate function $\mathbf{B}\Delta(y, \cdot)$ is an extension when*

$$\mathbf{B}\Delta(y, \cdot) = \Delta(y, \cdot) \tag{27}$$

on the vertices of the 0-1 unit cube under the mapping to $\mathbb{R}^p$: $i = \{1, \ldots, p\}$, $[u]^i = 1 - h^{\pi_i}(x) y^{\pi_i}$

Theorem 1. *If $l \in \mathcal{G}_+$, then $\mathbf{B_D}\Delta \geq \mathbf{S}\Delta$ over the unit cube given in Definition 10, and therefore $\mathbf{B_D}$ is closer to the convex closure of Δ than $\mathbf{S}$.*

Proof. By the definition of the Lovász hinge $\mathbf{L}$ [30], we know that for any modular function $\Delta_{\mathcal{S}}$ we have $\mathbf{L}\Delta_{\mathcal{S}} \geq \mathbf{S}\Delta_{\mathcal{S}}$ over the unit cube. As a result of Lemma 1 and Lemma 2,we have

$$\begin{aligned} \mathbf{B_D}\Delta &= \mathbf{S}\Delta_{\mathcal{G}} + \mathbf{L}\Delta_{\mathcal{S}} = \mathbf{S}(\Delta - \Delta_{\mathcal{S}}) + \mathbf{L}\Delta_{\mathcal{S}} \\ &\geq \mathbf{S}\Delta - \mathbf{S}\Delta_{\mathcal{S}} + \mathbf{S}\Delta_{\mathcal{S}} = \mathbf{S}\Delta. \end{aligned}$$

□

Theorem 2. *If $l \in \mathcal{S}$, then $\mathbf{B_D}\Delta = \mathbf{L}\Delta$*

Proof. For $l \in \mathcal{S}$, we construct $g^* = \mathbf{0}$, and $f^* = l$ is submodular. By Definition 6, $g^*(V)$ is minimum, so $g^* = \mathbf{D}l$. Then $\mathbf{B_D}\Delta = \mathbf{L}\Delta + \mathbf{S0} = \mathbf{L}\Delta$ □

Theorem 3. *$\mathbf{B_D}\Delta$ is convex for arbitrary Δ.*

Proof. By Definition 6, $\mathbf{B_D}\Delta$ is the sum of the two convex surrogates, which is a convex surrogate. □

Theorem 4. *$\mathbf{B_D}\Delta$ is an extension of Δ iff $\Delta_{\mathcal{S}}$ is non-negative.*

Proof. From [30, Proposition 1], $\mathbf{S}\Delta$ is an extension for any supermodular increasing Δ; $\mathbf{L}\Delta$ is an extension iff Δ is submodular and non-negative as in this case, $\mathbf{L}$ coincides with the Lovász extenstion [17]. By construction from Definition 6 we have $\Delta_{\mathcal{G}}$ and $\Delta_{\mathcal{S}}$ for $g \in \mathcal{G}_+$ and $f \in \mathcal{S}$, respectively. Thus Equation 27 holds for both $\mathbf{S}\Delta_{\mathcal{G}}$ and $\mathbf{L}\Delta_{\mathcal{S}}$ if $\Delta_{\mathcal{S}}$ is non-negative. Then $\mathbf{B_D}\Delta$ taking the sum of the two extensions, Equation 27 also holds for every vertex of the unit cube as $\Delta = \Delta_{\mathcal{G}} + \Delta_{\mathcal{S}}$, which means $\mathbf{B_D}$ is also an extension of Δ . □

Theorem 5. *The subgradient computation of $\mathbf{B_D}\Delta$ is polynomial time given polynomial time oracle access to f^* and g^*.*

Proof. Given f^* and g^* we know that the subgradient computation of $\mathbf{L}\Delta_{\mathcal{S}}$ and $\mathbf{S}\Delta_{\mathcal{G}}$ are each polynomial time. Thus taking the sum of the two is also polynomial time. □

4 Sørensen-Dice loss

The Sørensen-Dice criterion [5, 26] is a popular criterion for evaluating diverse prediction problems such as image segmentation [23] and language processing [22]. In this section, we introduce the Sørensen-Dice loss based on the Sørensen-Dice coefficient. We prove that the Sørensen-Dice loss is neither supermodular nor submodular, and we will show in the experimental results section that our novel convex surrogate can yield improved performance on this measure.

Definition 11 (Sørensen-Dice Loss)**.** *Denote $y \subseteq V$ is a set of positive labels, e.g. foreground pixels, the Sørensen-Dice loss on given a groundtruth y and a predicted output $\tilde{y}$ is defined as*

$$\Delta_D(y, \tilde{y}) = 1 - \frac{2|y \cap \tilde{y}|}{|y| + |\tilde{y}|}. \tag{28}$$

Proposition 6. *$\Delta_D(y, \tilde{y})$ is neither submodular nor supermoduler under the isomorphism $(y^*, \tilde{y}) \to A := \{i | y_i^* \neq \tilde{y}_i\}$, $\Delta_J(y^*, \tilde{y}) \cong l(A)$.*

We will use the diminishing returns definition of submodularity in Definition 3 to first prove the following lemma:

Lemma 3. *Δ_D restricted to false negatives is neither submodular nor supermoduler.*

Proof. With the notation $m := |y^*| > 0$, $p := |\tilde{y} \setminus y^*|$, and $n := |y^* \setminus \tilde{y}|$, we have that

$$\Delta_D(y, \tilde{y}) = 1 - \frac{2m - 2n}{2m - n + p} = \frac{n + p}{2m - n + p} \tag{29}$$

For a given groundtruth y i.e. m, we have if $B \subseteq A$, then $n_B \leq n_A$, and $p_B \leq p_A$.

Considering i is an extra false negative, we calculate the marginal gain on A and B respectively:

$$\begin{aligned}
&\Delta_D(A \cup \{i\}) - \Delta_D(A) \\
&= \frac{n_A + 1 + p_A}{2m - n_A - 1 + p_A} - \frac{n_A + p_A}{2m - n_A + p_A} &(30)\\
&= \frac{2m + 2p_A}{(2m - n_A + p_A - 1)(2m - n_A + p_A)} &(31)\\
&\Delta_D(B \cup \{i\}) - \Delta_D(B) \\
&= \frac{2m + 2p_B}{(2m - n_B + p_B - 1)(2m - n_B + p_B)}. &(32)
\end{aligned}$$

Numerically, we have following counter examples which prove that Δ_D restricted to false negatives is neither submodular nor supermodular. We set $m = 10$, $n_A = [1:8]$, $n_B = n_A - 1 \leq n_A$, $p_A = 8$, $p_B = 5 \leq p_A$, and we plot the values of Equation (31) and Equation 32 as a function of n_A. We can see from Figure 1 that there exists a cross point between these two plots, which indicates that submodularity (Definition 3) does not hold for Δ_D or its negative.

□

Lemma 3 implies Proposition 6 as the restriction of a submodular function is itself submodular.

5 Experimental Results

We demonstrate the correctness and feasibility of the proposed convex surrogate on experiments using Dice loss, as well as on a face classification problem from video sequences with a family of non-modular losses.

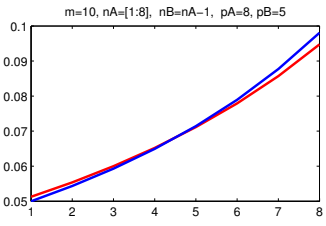


Figure 1: Plots of Eq. (31) (red) and Eq. 32 (blue) as a function of n_A. As these two plots cross, neither function bounds the other.

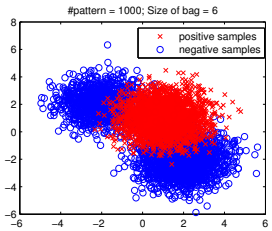


Figure 2: The data for the synthetic problem. The negative samples are drawn from a mixture of Gaussians.

5.1 Dice loss

We test the proposed surrogate on a binary set prediction problem. Two classes of 2-dimensional data are generated by different Gaussian mixtures as shown in Fig 2. We use the $\mathbf{B_D}$ during training time with the non-modular loss Δ_D to construct a convex surrogate. We compare it to slack rescaling $\mathbf{S}$ with an approximate optimization procedure based on greedy maximization. We additionally train an SVM (denoted 0-1 in the results table) for comparison. During test time, we evaluate with Δ_D and with Hamming loss to calculate the empirical error values as shown in Table 1.

We can see from the result that training Δ_D with $\mathbf{B_D}$ yields the best result while using Δ_D during test time. $\mathbf{B_D}$ performs better than $\mathbf{S}$ in both cases due to the failure of the approximate maximization procedure necessary to maintain computational feasibility [15].

5.2 Face classification in video sequences

We also evaluate the proposed convex surrogate operator on a real-world face track dataset [7, 8, 25]. The frames of the dataset are from the TV series "Buffy the Vampire Slayer". This dataset contains 1437 tracks and 27504 frames in total.

We focus on a binary classification task to recognize the leading role: "Buffy" is positive-labelled, "not Buffy" is negative-labelled. Example images are shown in Fig. 4. Each track is represented as a bag of frames, for which the size of the tracks varies from 1 frame to more than 100 frames, and each image is represented as a Fisher Vector Face descriptor of dimension 1937.

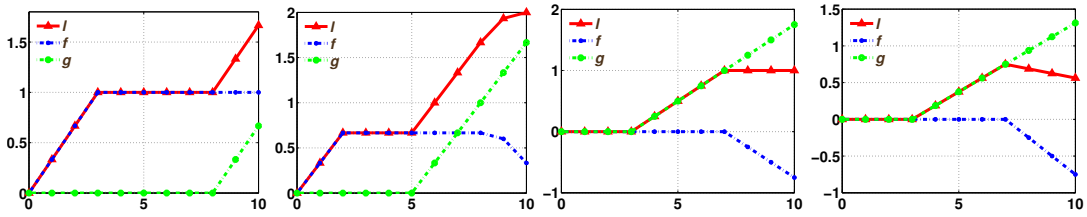
We have used different non-supermodular and non-submodular loss functions in our experiments as shown

$p=6$	Test	
	Δ_D	0-1
$\mathbf{B_D}$	$\mathbf{0.1121 \pm 0.0040}$	0.6027 ± 0.0125
0-1	0.1497 ± 0.0046	0.5370 ± 0.0114
S	0.3183 ± 0.0148	0.7313 ± 0.0209

Table 1: For the synthetic data experiment, the cross comparison of average loss values (with standard error) using different surrogate operations during training, and different evaluation functions during test time. Δ_D is the Dice loss as in Eq. (28).

	loss functions			
	Δ_1	Δ_2	Δ_3 (Δ_S negative)	Δ_4 (Δ_S negative)
$\mathbf{B_D}$	$\mathbf{0.194 \pm 0.006}$	$\mathbf{0.238 \pm 0.008}$	0.148 ± 0.005	0.108 ± 0.004
0-1	0.228 ± 0.007	0.284 ± 0.004	0.144 ± 0.004	0.107 ± 0.003
S	0.398 ± 0.015	0.243 ± 0.005	0.143 ± 0.006	0.106 ± 0.003

Table 2: For the face classification task, the cross comparison of average loss values (with standard error) using different surrogate operator and losses as in Equation (33) to Equation (36) during training, respectively. For the cases that the submodular component is non-negative, i.e. using Δ_1 and Δ_2, the lowest empirical error is achieved when using $\mathbf{B_D}$.



(a) Δ_1 (b) Δ_2 (c) Δ_3 (d) Δ_4

Figure 3: The plot of the four loss functions used in our experiments as in Equations (33) to (36). The x axis is the number of mispredictions for each track (we show here the loss functions corresponding to track length equal to 10 as an example), and the y axis is the value of loss function. The original losses are drawn in red; the supermodular components are drawn in green, and the submodular components in blue.

(a) (b) (c)

Figure 4: Examples of the face track images. Fig. 4(a) shows the "Buffy" role thus a positive-labelled image and Fig. 4(b) shows a negative-labelled image. An automated pipeline described in [7, 8, 25] was used for feature extraction (Fig. 4(c)).

in Equations 33 to 36:

$$\Delta_1(y, \tilde{y}) = \min(|\mathbf{I}|, |y|/3, |\mathbf{I}| - |y|/3) \quad (33)$$

$$\Delta_2(y, \tilde{y}) = \min(|\mathbf{I}|, |y|/4, |\mathbf{I}| - |y|/4, \alpha) \quad (34)$$

$$\Delta_3(y, \tilde{y}) = \min(\max(0, |\mathbf{I}| - |y|/3), |y|/3) \quad (35)$$

$$\Delta_4(y, \tilde{y}) = \min(\max(0, |\mathbf{I}| - |y|/3), \alpha) \quad (36)$$

$\mathbf{I} = \{i | y^i \neq \tilde{y}^i\}$ gives the set of incorrect prediction elements; α is a parameter that allows us to define the value of $l(V)$. Due to the fact that the size of the tracks varies widely, we further normalize the loss function with respect to the track size. We use $\alpha = 2$ for Δ_2 and $\alpha = 0.5$ for Δ_4 in the experiments.

As we can see explicitly in Fig. 3, no Δ is supermodular or submodular. Δ_1, Δ_2 and Δ_3 are increasing loss functions, while Δ_4 is non-increasing. For Δ_1 and Δ_2, we notice that the values of the set functions on a single element are non-zero i.e. $l_1(\{j\}) > 0$, $l_2(\{j\}) > 0$, $\forall j \in V$; while for the loss Δ_3 and Δ_1 these values are zero i.e. $l_3(\{j\}) = l_4(\{j\}) = 0$, $\forall j \in V$.

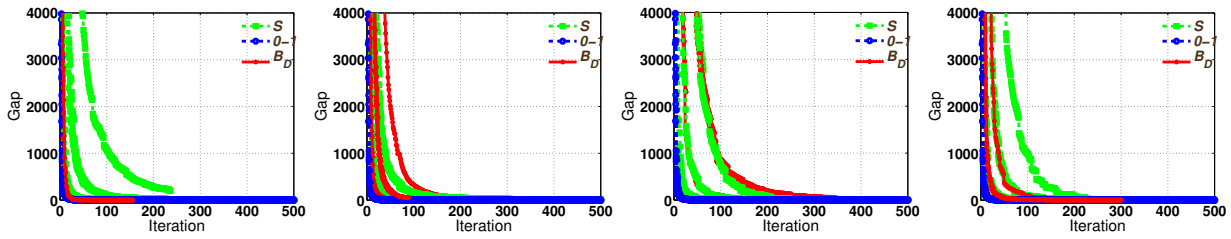
Fig. 3 shows the corresponding decomposition of each loss into the supermodular and submodular components as specified in Definition 6. We denote each loss function as $l_k = f_k + g_k$, for $k = \{1, 2, 3, 4\}$.

By construction, all supermodular g_k, for $k = \{1, 2, 3, 4\}$, are non-negative increasing. For the submodular component, f_1 is non-negative increasing, f_2 is non-negative and non-increasing, while f_3 and f_4 are both non-positive decreasing.

We compare different convex surrogates during training for these non-modular functions. And we additionally train on the Hamming loss (labelled 0-1) as a comparison. As training non-supermodular loss with slack rescaling is NP-hard, we have employed the simple application of the greedy approach as in [15].

10-fold-cross-validation has been carried out and we obtain an average performance and standard error as shown in Table 2.

From Table 2 we can see that when the submodular



(a) The gap using Δ_1 (b) The gap using Δ_2 (c) The gap using Δ_3 (d) The gap using Δ_4

Figure 5: The primal-dual gap as a function of the number of cutting-plane iterations using different convex surrogates for the four non-modular functions in Equations (33) to 36. The primal-dual gap from $\mathbf{B_D}$ is drawn in red; the gap from $\mathbf{S}$ is drawn in green, and gap from Hamming loss (labelled 0-1, and equivalent to a SVM) in blue. Our convex surrogate operator $\mathbf{B_D}$ can achieve a comparable convergence rate to an SVM, demonstrating that optimization is very fast in practice and the method scales well to large datasets.

	$p = 10$	$p = 50$	$p = 100$
$\mathbf{B_D}$	0.002 ± 0.000	0.018 ± 0.003	0.060 ± 0.008
$\mathbf{S}$	0.002 ± 0.000	0.016 ± 0.002	0.057 ± 0.002

Table 3: The comparison of the computation time (s) for one loss augmented inference.

component of the decomposition is non-negative, i.e. in the case of using Δ_1 and Δ_2, the lowest empirical error is achieved by using our convex surrogate operator $\mathbf{B_D}$.

Fig. 5 shows the primal-dual gap as a function of the cutting plane iterations for each experiment using different loss functions and different convex surrogate operators. We can see that in all cases, the convergence of $\mathbf{B_D}$ is at a rate comparable to an SVM, supporting the wide applicability and scalability of the convex surrogate. We have also compared the expected time of one loss augmented inference. Table 3 shows the comparison using Δ_1 with $\mathbf{B_D}$ and $\mathbf{S}$. As the cost per iteration is comparable to slack-rescaling, and the number of iterations to convergence is also comparable, there is consequently no computational disadvantage to using the proposed framework, while the statistical gains are significant.

6 Discussion and Conclusions

The experiments have demonstrated that the proposed convex surrogate is efficient, scalable, and reduces test time error for a range of loss functions, including the Sørensen-Dice loss, which is a popular evaluation metric in many problem domains. We see that slack rescaling with greedy inference can lead to poor performance for non-supermodular losses. This is especially apparent for the results of training with Δ_1, in which the test-time loss was approximately double that of the proposed method. Similarly, ignoring the loss function and simply training with 0-1 loss can lead to comparatively poor performance, e.g. Δ_1 and Δ_2. This clearly demonstrates the strengths of the proposed method for non-modular loss functions for which a decomposition with a non-negative submodular component is possible (Δ_1 and Δ_2, but not Δ_3 or Δ_4). The characterization and study of this family of loss functions is a promising avenue for future research, with implications likely to extend beyond empirical risk minimization with non-modular losses as considered in this paper. The primal-dual convergence results empirically demonstrate that the loss function is feasible to apply in practice, even on a dataset consisting of tens of thousands of video frames. The convex surrogate is directly amenable to other optimization techniques, such as stochastic gradient descent [3], or Frank-Wolfe approaches [16], as well as alternate function classes including neural networks.

In this work, we have introduced a novel convex surrogate for general non-modular loss functions. We have defined a decomposition for an arbitrary loss function into a supermodular non-negative function and a submodular function. We have proved both the existence and the uniqueness of this decomposition. Based on this decomposition, we have proposed a novel convex surrogate operator taking the sum of two convex surrogates that separately bound the supermodular component and the submodular component using slack-rescaling and the Lovász hinge, respectively. We have demonstrated that our new operator is a tighter approximation to the convex closure of the loss function than slack rescaling, that it generalizes the Lovász hinge, and is convex, piecewise linear, an extension of the loss function, and for which subgradient computation is polynomial time. Open-source code of ℓ_2 regularized risk minimization with this operator is available for download from `https://github.com/yjq8812/aistats2016`.

Acknowledgements

This work is partially funded by Internal Funds KU Leuven, ERC Grant 259112, and FP7-MC-CIG 334380. The first author is supported by a fellowship from the China Scholarship Council.

References

[1] F. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning*, 6(2-3):145–373, 2013.

[2] M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In D. Forsyth, P. Torr, and A. Zisserman, editors, *European Conference on Computer Vision*, volume 5302 of *Lecture Notes in Computer Science*, pages 2–15. Springer, 2008.

[3] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, pages 161–168, 2008.

[4] W. Cheng, E. Hüllermeier, and K. J. Dembczynski. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the International Conference on Machine Learning*, pages 279–286, 2010.

[5] L. R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, July 1945.

[6] J. R. Doppa, J. Yu, C. Ma, A. Fern, and P. Tadepalli. HC-search for multi-label prediction: An empirical study. In *Proceedings of AAAI Conference on Artificial Intelligence*, 2014.

[7] M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is... Buffy" – automatic naming of characters in TV video. In *Proceedings of the British Machine Vision Conference*, 2006.

[8] M. Everingham, J. Sivic, and A. Zisserman. Taking the bite out of automatic naming of characters in TV video. *Image and Vision Computing*, 27(5):545–559, 2009.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[10] T. Finley and T. Joachims. Training structural SVMs when exact inference is intractable. In *Proceedings of the 25th International Conference on Machine Learning*, pages 304–311, 2008.

[11] S. Fujishige. *Submodular functions and optimization*. Elsevier, 2005.

[12] J. Gillenwater, R. Iyer, B. Lusch, R. Kidambi, and J. Bilmes. Submodular Hamming metrics. In *Neural Information Processing Society (NIPS)*, Montreal, Canada, December 2015.

[13] R. Iyer and J. Bilmes. Algorithms for approximate minimization of the difference between submodular functions, with applications. In *Uncertainty in Artificial Intelligence (UAI)*, 2012.

[14] T. Joachims, T. Finley, and C.-N. Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59, 2009.

[15] A. Krause and D. Golovin. Submodular function maximization. In L. Bordeaux, Y. Hamadi, and P. Kohli, editors, *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press, 2014.

[16] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-coordinate Frank-Wolfe optimization for structural SVMs. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, pages 53–61, 2013.

[17] L. Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pages 235–257. Springer, 1983.

[18] O. L. Mangasarian. Uniqueness of solution in linear programming. *Linear Algebra and its Applications*, 25(0):151–162, 1979.

[19] M. Narasimhan and J. Bilmes. A submodular-supermodular procedure with applications to discriminative structure learning. In *Uncertainty in Artificial Intelligence (UAI)*, 2005.

[20] S. Nowozin. Optimal decisions from probabilistic models: The intersection-over-union case. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[21] J. Petterson and T. S. Caetano. Submodular multi-label learning. In *Advances in Neural Information Processing Systems*, pages 1512–1520, 2011.

[22] P. Rychlỳ. A lexicographer-friendly association score. In *Proceedings of Recent Advances in Slavonic Natural Language Processing*, 2008.

[23] M. R. Sabuncu, B. T. Yeo, K. Van Leemput, B. Fischl, and P. Golland. A generative model for

image segmentation based on label fusion. *Medical Imaging, IEEE Transactions on*, 29(10):1714–1729, 2010.

[24] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003.

[25] J. Sivic, M. Everingham, and A. Zisserman. "Who are you?" – learning person specific classifiers from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[26] T. Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biol. Skr.*, 5:1–34, 1948.

[27] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 25–32. MIT Press, 2004.

[28] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(9):1453–1484, 2005.

[29] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[30] J. Yu and M. B. Blaschko. Learning submodular losses with the Lovász hinge. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Journal of Machine Learning Research: W&CP*, pages 1623–1631, Lille, France, 2015.